IOM

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/806,490	04/12/2001	Maciej Kubiczek	108347-00005	6264

4372 7590 02/24/2004

ARENT FOX KINTNER PLOTKIN & KAHN
1050 CONNECTICUT AVENUE, N.W.
SUITE 400
WASHINGTON, DC 20036

EXAMINER	
GERSTL, SHANE F	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 02/24/2004 4

Please find below and/or attached an Office communication concerning this application or proceeding.

IOM

Office Action Summary	Application No.	Applicant(s)	
	09/806,490	KUBICZEK ET AL.	
	Examiner	**Art Unit**	
	Shane F Gerstl	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *4/12/01*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *16-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *16-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *4/12/01* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☒ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *4/12/01*.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

1. Claims 16-30 have been examined.

Papers Received

2. Receipt is acknowledged of the preliminary amendment paper submitted, where the paper has been placed of record in the file.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: Virtual Machine Processing System That Executes Predefined 8-bit Virtual Machine Instructions and Fetches Subroutines of Microcode For User-defined 8-bit Virtual Machine Instructions.

4. This application does not contain an abstract of the disclosure as required by 37 CFR 1.72(b). An abstract on a separate sheet is required.

5. The following guidelines illustrate the preferred layout for the specification of a utility application. These guidelines are suggested for the applicant's use.

Arrangement of the Specification

As provided in 37 CFR 1.77(b), the specification of a utility application should include the following sections in order. Each of the lettered items should appear in upper case, without underlining or bold type, as a section heading. If no text follows the section heading, the phrase "Not Applicable" should follow the section heading:

(a) TITLE OF THE INVENTION.
(b) CROSS-REFERENCE TO RELATED APPLICATIONS.
(c) STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT.
(d) INCORPORATION-BY-REFERENCE OF MATERIAL SUBMITTED ON A COMPACT DISC (See 37 CFR 1.52(e)(5) and MPEP 608.05. Computer program listings (37 CFR 1.96(c)), "Sequence Listings" (37 CFR 1.821(c)),

and tables having more than 50 pages of text are permitted to be submitted on compact discs.) or
REFERENCE TO A "MICROFICHE APPENDIX" (See MPEP § 608.05(a). "Microfiche Appendices" were accepted by the Office until March 1, 2001.)

(e) BACKGROUND OF THE INVENTION.
   (1) Field of the Invention.
   (2) Description of Related Art including information disclosed under 37 CFR 1.97 and 1.98.
(f) BRIEF SUMMARY OF THE INVENTION.
(g) BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING(S).
(h) DETAILED DESCRIPTION OF THE INVENTION.
(i) CLAIM OR CLAIMS (commencing on a separate sheet).
(j) ABSTRACT OF THE DISCLOSURE (commencing on a separate sheet).
(k) SEQUENCE LISTING (See MPEP § 2424 and 37 CFR 1.821-1.825. A "Sequence Listing" is required on paper if the application discloses a nucleotide or amino acid sequence as defined in 37 CFR 1.821(a) and if the required "Sequence Listing" is not submitted as an electronic document on compact disc).

Drawings

6. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference sign(s) not mentioned in the description: 107, 251, and 302. A proposed drawing correction, corrected drawings, or amendment to the specification to add the reference sign(s) in the description, are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

7. Claims 19 and 21 are objected to because of the following informalities: these dependent claims are not grouped together with the claims that they refer to as specified in 37 CFR 1.75 (g).

8. Claims 16 and 30 are objected to because of the following informalities: There are extra characters ("--") on the same lines as these claims. Though the examiner is

taking the characters to simply be beginning and end points in the claim. There is some uncertainty on whether additional claim language is meant to be placed in lieu of these characters.

9. Claim 17 is objected to because of the following informalities: the last line of the claim mentions "check for the presence of a bit in that bit position". It is inherent that a bit position contains a bit. The examiner is taking the claim to mean "check for the presence of a predefined value of the bit" to show that a certain value of the bit is checked for rather than is existence as indicated in the spec.

10. Claim 19 is objected to because of the following informalities: the last line of the claim speaks of "the code". It is not totally clear whether this code is the same as the instruction code. The examiner is taking this code to be the same as the instruction code (based on the specification and claim 20's limitation of the bit shifted instruction) and suggests that the applicant change the claim to read "the instruction code".

11. Claim 22 is objected to because of the following informalities: the claim mentions "...pointing to the next instruction in the instruction memory when execution..." which is somewhat inconsistent with the specification where the next instruction is pointed to for next execution when the execution is completed. The examiner is taking the claim to mean just that, "...pointing to the next instruction in the instruction memory to be executed when execution..."

12. Claim 30 is objected to because the examiner wishes to be sure that the applicant desires to limit the claim using "consisting of" language. MPEP 2111.03 shows that the term "consisting of" is closed language and excludes any element or

step not specified in the claim. In other words, the invention is limited to only that which is explicitly stated in the claims and nothing more or less. There is no error in using such language; the examiner is simply making sure that the applicant wishes to limit the scope of the invention in such a manner.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

13. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

14. Claims 16-30 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

15. Claim 16 recites the limitation "the sequence" in line 1 of page 2. There is insufficient antecedent basis for this limitation in the claim. Only the fact that there are 8-bit microprocessor instructions has been defined. There has been no prior reference to a sequence of 8-bit instructions, only that there are 8-bit instructions. The examiner is taking the claim to mean, “a sequence” of 8-bit microprocessor instructions.

16. Claim 16 recites the limitation "an instruction" in line 4 of page 2. There is insufficient antecedent basis for this limitation in the claim. It is uncertain whether “an instruction” means a different instruction than one of the already defined stored instructions. The examiner is taking this instance of “an instruction” to mean “each instruction” in order to refer to the already defined instructions and fit the context better.

17. Claim 16 recites the limitation "an instruction" in lines 7, 8, and 10 of page 2. There is insufficient antecedent basis for this limitation in the claim. It is uncertain whether "an instruction" means a different instruction than one of the already defined stored instructions. The examiner is taking these instances of "an instruction" to mean "one of the stored instructions" in order to refer to the already defined instructions and fit the context better.

18. Claim 16 recites the limitation "a subroutine" in line 10 of page 2. There is insufficient antecedent basis for this limitation in the claim. The limitation "the generated address" in line 11 is used to refer back to the address generated previously by the means for generating an address. There is already defined a subroutine for the case when an address is generated (claim 16). Therefore, it is unclear whether this subroutine or a new one is meant to be used. The examiner is taking the claim to mean "the subroutine" in order to better fit the context.

19. Claim 23 recites the limitations "(the) data memory" and "(the) program counter" in the claim. There is insufficient antecedent basis for this limitation in the claim. The first line of the claim introduces the series of limitations with the article "the" meaning that each limitation must have been previously defined. However, a data memory and a program counter have not yet been defined. The examiner is taking the claim to mean "...wherein the central processing, the instruction memory, a data memory, the hardware stack, and a program counter..." to remedy the problem.

20. Claim 27 recites the limitation "the fast call bit" in the third line of the claim. There is insufficient antecedent basis for this limitation in the claim. No such bit has been defined so the examiner will take the claim to mean “a fast call bit”.

21. Claim 28 recites the limitation "said other dedicated bit" in the claim. There is insufficient antecedent basis for this limitation in the claim. It is unclear whether this is some other bit or the same predefined “another bit” from the parent. The examiner is taking the claim to mean “said another dedicated bit” based on the specification.

22. Claim 30 recites the limitations "the program counter register" in line 1 of page 5 and “the 8-bit bytecode” in lines 2-3 of page 5. There is insufficient antecedent basis for this limitation in the claim. Neither of these limitations have been previously defined so the examiner is taking the claim to mean “a program counter register” and “an 8-bit bytecode”.

Claim Rejections - 35 USC § 102

23. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

24. Claims 16-18, 21-27, and 29 are rejected under 35 U.S.C. 102(e) as being anticipated by Lindwer (6,298,434).

25. In regard to claim 1, Lindwer discloses a microprocessor system for executing Virtual Machine bytecodes which have been translated into respective 8-bit microprocessor instructions which correspond either to a fixed and predefined operation or to a user defined operation, the system comprising:

a. A central processing unit (figure 1, element 112); This element is shown in column 3, line 63 to be a processor core, or a central processing unit.

b. An instruction memory for storing the sequence of 8-bit microprocessor instructions (figure 1,element 120); Column 3, lines 62-63 show that this element is an instruction memory. Column 4, lines 7-9 show that virtual machine instructions are fetched from the instruction memory and thus stored there. Column 4, lines 42-43 show that one such virtual instruction of use is a Java bytecode. A bytecode inherently consists of a byte and is thus 8-bits long. Therefore, the instruction memory that stores these virtual machine bytecodes stores sequences of 8-bit instructions for the microprocessor.

c. Means for fetching each stored instruction in turn and for analyzing each instruction to determine whether an instruction corresponds either to a fixed and predefined operation or to a user defined operation; Column 4, lines 7-9 shows that the virtual machine instructions stored in the instruction memory are fetched when required, and thus in turn. Column 8, lines 42-45 show that there are special virtual machine instructions among these fetched virtual machine instructions. Since the special virtual machine instructions supply a jump to subroutine of instructions (column 8, lines 42-48) and regular virtual machine

instructions are simply converted and sent to the processor core for execution (column 6, lines 48-64), there must be means for analyzing the virtual machine instructions to determine what type they are. This is illustrated in figure 4 by decision block 42. The special virtual machine instructions are user-defined instructions because they point to a subroutine of native instructions in the instruction memory, which are set there by the compiler, or user-defined. The other virtual machine instructions are predefined and fixed because a hardware converter is used to translate the instructions directly.

d. Means for generating an address corresponding to the location of a subroutine if an instruction corresponds to a user defined operation; As shown above, column 8, lines 42-48 show that for a user defined operation, a subroutine is called based on a jump address of the instruction. It is inherent that this address is generated.

e. Wherein, in the event that an instruction corresponds to a fixed and predefined operation, the instruction is passed to the central processing unit for execution and, in the event that an instruction corresponds to a user defined operation, a subroutine corresponding to the instruction is called using the generated address. As shown above, the fixed and predefined operations are converted and sent to the processor core. Also as shown above, for the user defined operations, a subroutine corresponding to the instruction is called using the generated target address.

26. In regard to claim 17, Lindwer discloses a microprocessor system according to claim 16, wherein instructions corresponding to fixed and predefined operations are distinguished from instructions corresponding to user defined operations by a bit in a predefined bit position of the instruction code, and the means for analyzing each stored instruction is arranged to check for the presence of a predefined value in that bit position. It is inherent that the predefined and user defined instructions are distinguished by one or more predefined bits in the instruction code, which is then checked to identify the instruction type.

27. In regard to claim 18, Lindwer discloses a microprocessor system according to claim 16, wherein the microprocessor system comprises a data memory arranged in use to store code defining said subroutines. Column 8, lines 45-48 show that the subroutine code is stored in an instruction memory. Since this memory stores instructions including the subroutines, which consist of data, the memory is a data memory.

28. In regard to claim 21, Lindwer discloses a microprocessor system according to claim 18, wherein the instruction memory is arranged to hold 8-bit wide instructions, while the data memory is arranged to hold 32-bit data values. As shown above, the instruction memory holds data and is thus also a data memory. Also above is shown that the memory stores 8-bit wide instructions. Column 6, lines 11-15 show that virtual machine instructions are converted into instructions for a 32-bit MIPS processor. This means that the MIPS processor uses 32-bit instructions (as shown in column 7, lines 26-28, where 4-bytes is 32-bits) and operand data. Column 5, lines 10-11 show that the

memory also contains native instructions (instructions for the MIPS processor) and thus holds the 32-bit MIPS instructions, which are data values.

29. In regard to claim 22, Lindwer discloses a microprocessor system according to claim 16, and comprising a hardware stack arranged in use to store a return address when a subroutine is entered, the return address pointing to the next instruction in the instruction memory to be executed when execution of the subroutine is completed. Column 10, lines 12-17 show that the return address for a subroutine is stored in a register, which has been mapped there from a virtual location in the case of the virtual jump to a subroutine. Column 7, lines 38-49 show that the registers are mapped to the stack and thus the return address is stored on a hardware stack. Column 8, lines 57-60 show that on return from a subroutine processing of native instructions (which are fetched from memory as shown in column 5, lines 10-11) is resumed. The use of the word resumed means that the processing picks up where it left off to execute the subroutine. This is the next native instruction and thus the next instruction in the instruction memory is pointed to by the return address.

30. In regard to claim 23, Lindwer discloses a microprocessor system according to claim 22, wherein the central processing unit, the instruction memory, a data memory, the hardware stack, and a program counter are all coupled to a common bus. Since the instruction memory holds data in the form of instructions, it is also a data memory. Figure 1A shows that the central processing unit (element 112), the instruction memory (120), and a data memory (120) are all coupled to a common data bus (140). Column 4, lines 28-37 show that the processor core has an instruction and data port connected

to a program counter and registers and that the processor core is connected to memory (via the instruction and data port). Thus the program counter and registers are coupled to the same common data bus. Finally, since, as shown above, the registers are mapped to the stack, the hardware stack is also coupled to the common data bus.

31. In regard to claim 24, Lindwer discloses a microprocessor system according to claim 23, wherein the central processing unit, instruction memory, data memory, hardware stack, program counter, and common data bus are integrated onto a single chip. Column 4, lines 19-20 show that the entire system of mention, 110 of figure 1, is a single entity processor. The standard IEEE definition for the term "microprocessor" is shown to be on a single integrated circuit (chip). The references of this definition also show that a processor is synonymous with a microprocessor and thus the invention of Lindwer is integrated onto a single chip.

32. In regard to claim 25, Lindwer discloses a microprocessor system according to claim 16, wherein the central processing unit contains an arithmetic logic unit and a data stack, and the top two elements of the data stack are connected to the inputs of the arithmetic logic unit and the output of the arithmetic logic unit is connected to an internal data bus. Column 6, lines 7-11, show an example of an add operation that necessitates an arithmetic logic unit that uses a data stack. It is shown here that the operation take the top two values from the stack and adds them so the inputs of the arithmetic logic unit must be connected to the top two elements of the stack. Since the result of the addition is then pushed back onto the stack, also as shown here, it is inherent that there is an internal data bus to feed the stack the result from the arithmetic logic units output.

33. In regard to claim 26, Lindwer discloses a microprocessor system according to claim 25, wherein the top three elements of the data stack contain special purpose circuits, which enable the execution of seven primitive stack operations directly in hardware. Column 7, lines 38-42, show that the registers are mapped to the stack (including the top three elements) and in fact is its own register (data) stack. Special purpose circuits are required to enable this mapping. This mapping means that each operation that uses a register also uses the stack. Column 6, lines 11-15 show that the native processor is a MIPS processor. The textbook listing of MIPS instructions provided shows that there are seven and in fact more than seven primitive operations that use registers (denoted with a '$') and thus the register stack for execution in hardware.

34. In regard to claim 27, Lindwer discloses a microprocessor system according to claim 16, and comprising means for recognizing a fast return instruction folded with a regular instruction, by utilizing circuitry which decodes the fast call bit in an 8-bit bytecode and another dedicated bit or bits in the 8-bit bytecode. Column 8, lines 57-60 shows that when a return from a subroutine is detected native (regular) instructions resume issuing and thus since upon recognizing this return, regular instructions begin issuing again, the return and regular instructions are folded. This same paragraph shows that this return is from a subroutine called by a special 8-bit virtual instruction or user-defined instruction. Since as shown above, there is a set of bits that are decoded to identify this call, these bits can be called call bits or even fast call bits (which includes the another dedicated bit). It is inherent that the same circuitry that decoded these fast

call bits to initiate the subroutine decodes these bits again in order to monitor the return from the subroutine.

35. In regard to claim 29, Lindwer discloses a microprocessor system according to claim 16, wherein said Virtual Machine bytecodes are Java bytecodes as shown above.

Claim Rejections - 35 USC § 103

36. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

37. Claims 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lindwer in view of Evoy (5,937,193).

38. In regard to claim 19,

a. Lindwer discloses a microprocessor system according to claim 17,

b. Lindwer does not disclose wherein said distinguishing bit is the most significant bit of the instruction code, and the generating means is arranged to shift the instruction code to the left by one or more bits. Lindwer discloses that a special jump to subroutine instruction is used but does not disclose how the target address is generated.

c. Evoy shows on column 10, lines 45-57, that an address is generated for locating a section of native code (subroutine) using a Java bytecode (instruction code). This generating is done using a value obtained by shifting the bytecode left by two bits. Evoy shows in column 9, lines 38-41 that the shifting is done in

order to convert the platform-independent code (Java virtual machine code) into a 32-bit offset. Column 6, lines 11-15 of Lindwer show that virtual machine instructions are converted into instructions for a 32-bit MIPS processor. Thus, a 32-bit offset and address is needed to fetch native instruction subroutines from memory. Also, by storing the distinguished bit as the most significant bit of the instruction code, a conventional location is established for this bit or bits.

d. This ability to address the processor memory by using a simple shift and sum as taught would have motivated one of ordinary skill in the art to modify the design of Lindwer to incorporate the address generation design of Evoy and the ability to have a conventional location for the distinguished bit would have motivate done of ordinary skill in the art to store the distinguished bit as the most significant bit.

It would have been obvious to one of ordinary skill in the art to modify the design of Lindwer to incorporate the target address generation technique taught by Evoy so that the target address for a 32-bit memory may be generated in a simple manner and the examiner is taking official notice that it would have been obvious to modify the design of Lindwer to store the distinguished bit as the most significant bit so that a conventional location is established.

39. In regard to claim 20,

a. Lindwer in view of Evoy discloses a microprocessor system according to claim 19, and comprising a program counter register. Column 4, lines 32-33 of Lindwer shows that a program counter is an instruction register and column 10,

lines 45-51 of Evoy shows that the instruction code is loaded into an instruction register (program counter register).

b. Lindwer in view of Evoy does not disclose that the program counter register is arranged to load the *shifted* instruction code.

c. By storing the shifted instruction code in the program counter register instead of the non-shifted code, the shift calculation is already completed and stored and does not have to be calculated again when needed, thus saving time.

d. This time saved when the shifted data would normally be calculated would have motivated one of ordinary skill in the art to modify the design of Lindwer in view of Evoy to store the shifted instruction code in the program counter register.

The examiner is taking official notice that it would have been obvious to one of ordinary skill in the art to modify the design of Lindwer in view of Evoy to store the shifted instruction code in the program counter register instead of the non-shifted code so that time may be saved when the shifted data is needed.

40. Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lindwer.

41. In regard to claim 28,

a. Lindwer discloses a microprocessor system according to claim 27,

b. Lindwer does not disclose wherein said another dedicated bit is the second most significant bit in the 8-bit bytecode.

c. By putting the fast call bits as the most significant bits (or the fast call bit as the most significant bit and another dedicated bit as the second most

significant bit) of the bytecode, a conventional location is established to simplify finding these bits for decoding and recognition.

d. Establishing a conventional location for the fast call bits in order to simplify design would have motivated one of ordinary skill in the art to modify the design of Lindwer to hold these bits in a conventional location.

The examiner is taking official notice that It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Lindwer to hold the fast call bit and another bit or bits in the most significant bits of the instruction code so that these bits are stored in a conventional location.

42. Claim 30 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lindwer in view of Lee (4,937,783) and further in view of Evoy.

43. In regard to claim 30,

a. Lindwer discloses a microprocessor system, *comprising*

i. a central processing unit (figure 1, Proc);

ii. 32-bit wide data memory; Column 3, lines 62-63 shows an instruction memory. Since instructions are data, this is also a data memory. Column 5, lines 10-11 shows that this memory contains native instructions. Column 6, lines 11-14 shows that these native instructions are for a MIPS machine and thus are 32-bit instructions and thus the data memory is 32-bits wide.

iii. and a hardware stack connected via an internal bus to a program counter register, Column 7, lines 38-40 show that the machine is stack

oriented and does have a hardware stack. Column 8, lines 65-67 shows a program counter being incremented. Column 11, lines 3-5 show that the stack holds computational values. Therefore, since the program counter needs to have computation done (such as the incrementing), it is inherently connected to the stack.

iv. together with an instruction decode unit which includes a circuit for detecting the presence of a distinguished bit in an 8-bit bytecode; Since the machine is a MIPS machine it inherently has an instruction decode unit. Column 8, lines 42-45 show that there are special virtual machine instructions among the fetched virtual machine instructions. Since the special virtual machine instructions supply a jump to subroutine of instructions (column 8, lines 42-48) and regular virtual machine instructions are simply converted and sent to the processor core for execution (column 6, lines 48-64), there must be means for analyzing the virtual machine instructions to determine what type they are. This inherently means that there is a distinguished bit or bits to differentiate between the two types. Column 6, lines 8-11 show that Java bytecodes (8-bits) are the virtual instructions.

b. Lindwer does not disclose the microprocessor system comprising:

i. 8-bit wide instruction memory,

ii. together with a circuit for loading the remaining bits of the bytecode shifted left by a number of bits into the microprocessor's program counter

register, while at the same time storing the current value of the program counter register on the aforementioned stack.

Lindwer also does not disclose the microprocessor system *consisting of* the above limitations where the stated limitations are the only elements in existence within the system.

c. Lee has disclosed the use of an 8-bit wide instruction memory as shown in column 4, lines 3-5. This would allow for the design of Lindwer to avoid wasting space by storing 8-bit instructions in a 32-bit instruction. Evoy has disclosed a virtual machine microprocessor system together with a circuit for loading the remaining bits of the bytecode shifted left by a number of bits into the microprocessor's program counter register, while at the same time storing the current value of the program counter register on the aforementioned stack. Column 4, lines 32-33 of Lindwer shows that a program counter is an instruction register and column 10, lines 45-51 of Evoy shows that the instruction code is loaded into an instruction register (program counter register). Since computation (shifting) would be done to the value of the program counter, this value would be stored on the stack, as is shown to be done for computations in the design of Lindwer above. Evoy shows in column 9, lines 38-41 that the shifting is done in order to convert the platform-independent code (Java virtual machine code) into a 32-bit offset. Column 6, lines 11-15 of Lindwer show that virtual machine instructions are converted into instructions for a 32-bit MIPS processor. Thus, a 32-bit offset and address is needed to fetch native instruction subroutines from

memory. If one takes away all functionality and elements other than the above limitations away from the disclosure of Lindwer in view of Lee and further in view of Evoy, these claim limitations are left to stand alone in element and functionality. This leaves a machine with a low area and cost that functions to meet the limitations.

d. The ability to address the processor memory by using a simple shift and sum as taught would have motivated one of ordinary skill in the art to modify the design of Lindwer to incorporate the address generation design of Evoy. The ability to avoid wasting storage space for instructions would have motivated one of ordinary skill in the art to modify the design of Lindwer to use an 8-bit instruction memory as taught by Lee. Lower cost and circuit area would have motivated one of ordinary skill in the art at the time of invention to modify the design of Lindwer in view of Lee and further in view of Evoy to eliminate extra elements and functionality that are not needed.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Lindwer to use an 8-bit instruction memory as taught by Lee, and the address generation scheme taught by Evoy so that memory space is not wasted and simple address generation may be used, and the examiner is taking official notice that it would have been obvious to take away extra functionality and elements from the disclosure of Lindwer in view of Lee and further in view of Evoy so that a cheaper and smaller design may be used.

Conclusion

44. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

45. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The following patents have been cited to further show the art with respect to virtual machine processing in general.

US Pat No 5,898,850 to Dickol shows a Java bytecode processor that generates target addresses by shifting the opcode bits of the bytecode.

US Pat No 5,875,336 to Dickol discloses a microprocessor system that translates Java bytecodes into native instructions and for more complex instructions fetches a subroutine.

US Pat No. 6,003,038 to Chen illustrates a Java processor that directly executes some Java instructions and fetches microcode instructions for others.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
February 20, 2004

Eddie Chan

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100